Relational Database [RDBMS]

- Arpita Shah

Database Management System (DBMS)

- Collection of components that support data acquisition[Process data are received by the automation system, either locally from the controllers in centralized control systems or distributed systems or remotely from server] dissemination [release to users of information obtained through a statistical activity] , storage, maintenance, retrieval, and formatting
- Product variations

(1) Enterprise DBMSs

Oracle: dominates in Unix; strong in Windows

SQL Server: strong in Windows

DB2: strong in MVS and VM environments

Teradata: usage as a data warehouse platform

Amazon Web Services

SAP Sybase: possible challenge to Oracle

Significant open source DBMSs: MySQL, PostgreSQL, MongoDB, MariaDB, SQLite, Cassandra

Cloud-based and NoSQL: rapidly evolving

(2) Desktop DBMSs
Access: dominates
LibreOffice Base, Open Office Base, FileMaker Pro
(3) Embedded DBMSs

- Major part of information technology infrastructure

Database Management System

A **database management system (DBMS)** is a software that controls the storage, organization, and retrieval of data. Typically, a DBMS has the following elements:

- Kernel code

  This code manages memory and storage for the DBMS.
- Repository of metadata

  This repository is usually called a **data dictionary**.
- Query language

  This language enables applications to access the data.

Database solutions into two groups

1. Relational Database:

Microsoft SQL Server, Oracle Database, MySQL, and IBM DB2.

RDBMS's are mostly used in large enterprise scenarios

2. Non-relational Database: (NoSQL databases)

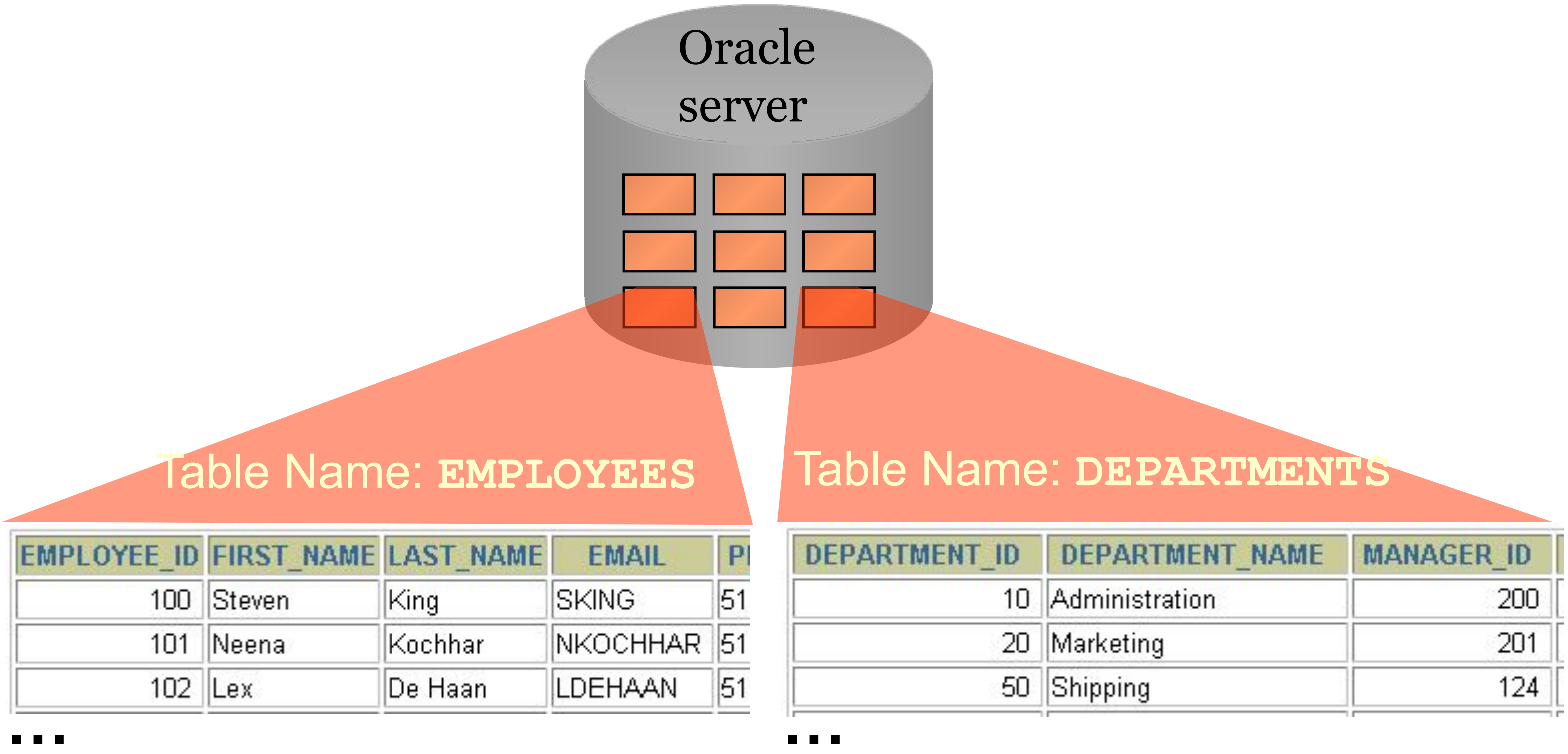
MongoDB, DocumentDB, Cassandra, CouchDB, Pnuts/Sherpa, HBase, Redis, Neo4,BigTable, Dynamo

Relational Database Concept

- The relational model consists of the following:
  - Collection of objects or relations
  - Set of operators to act on the relations
  - Data integrity for accuracy and consistency

A relational database is a collection of relations or two-dimensional tables

A relational database can contain one or many tables. A *table* is the basic storage structure of an RDBMS. A table holds all the data necessary about something in the real world, such as employees, departments, invoices, or customers.



Oracle server

Table Name: **EMPLOYEES**

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PI
100	Steven	King	SKING	51
101	Neena	Kochhar	NKOCHHAR	51
102	Lex	De Haan	LDEHAAN	51

…

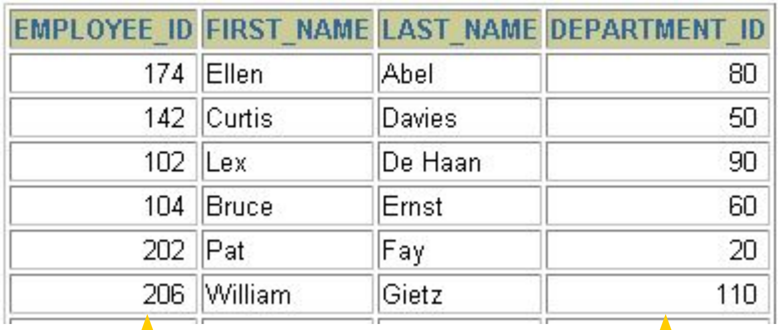
Table Name: **DEPARTMENTS**

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID
10	Administration	200
20	Marketing	201
50	Shipping	124

…

Relating Multiple Tables

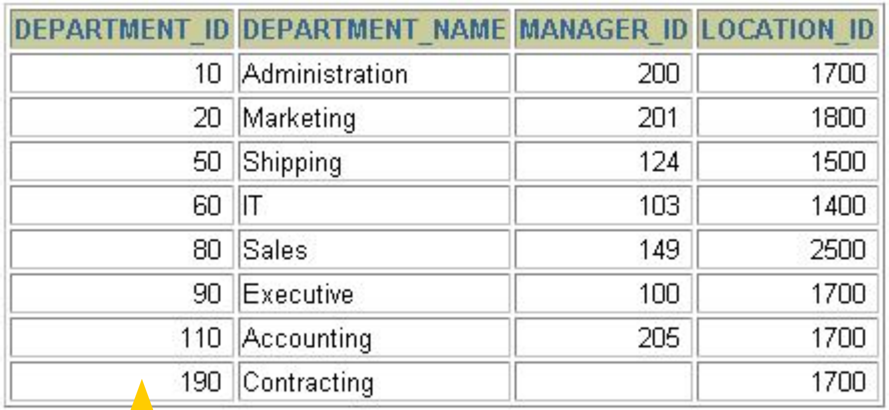
- Each row of data in a table is uniquely identified by a primary key (PK).
- You can logically relate data from multiple tables using foreign keys (FK).



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
174	Ellen	Abel	80
142	Curtis	Davies	50
102	Lex	De Haan	90
104	Bruce	Ernst	60
202	Pat	Fay	20
206	William	Gietz	110

…

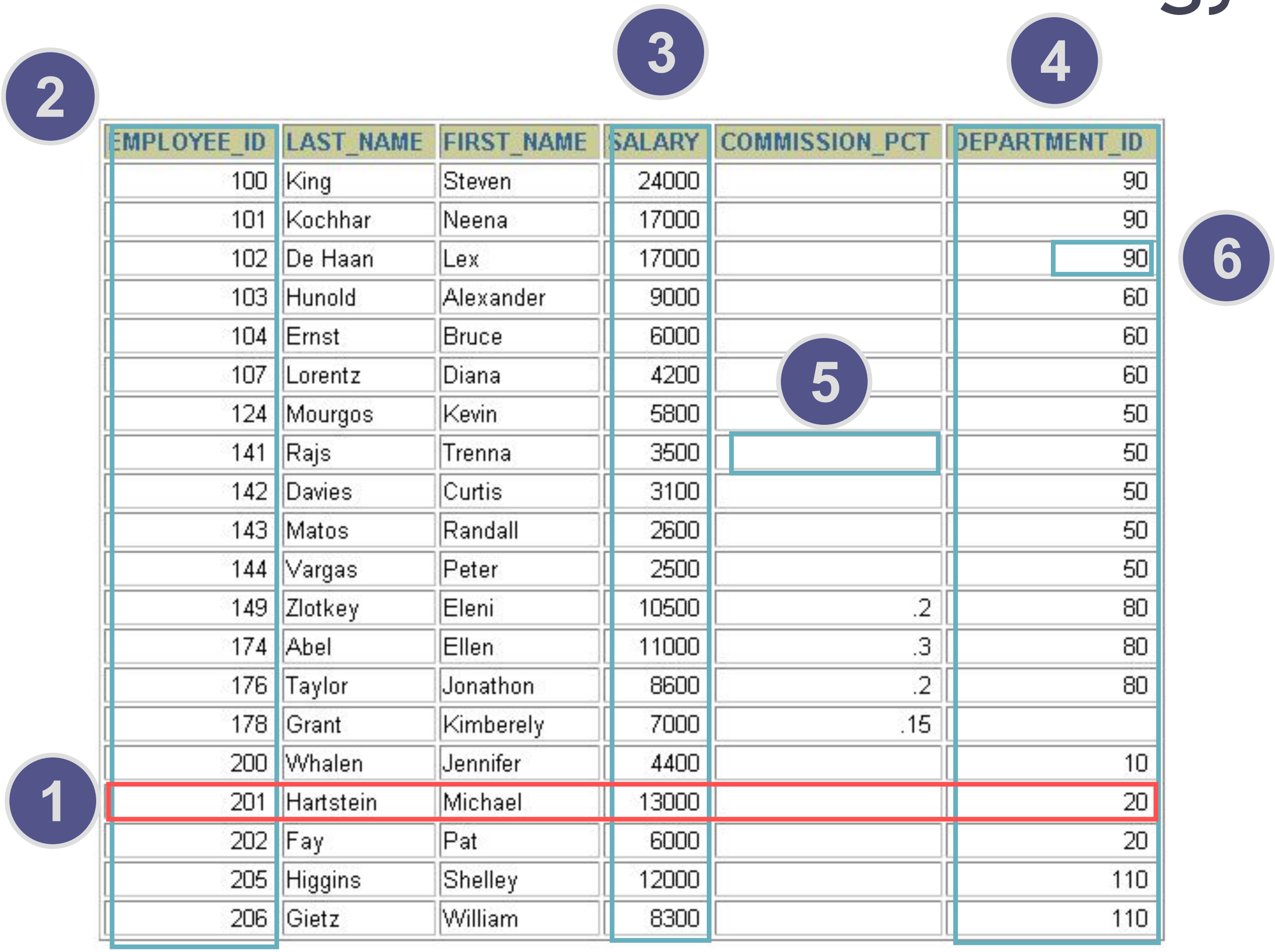
Table Name: **DEPARTMENTS**



DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

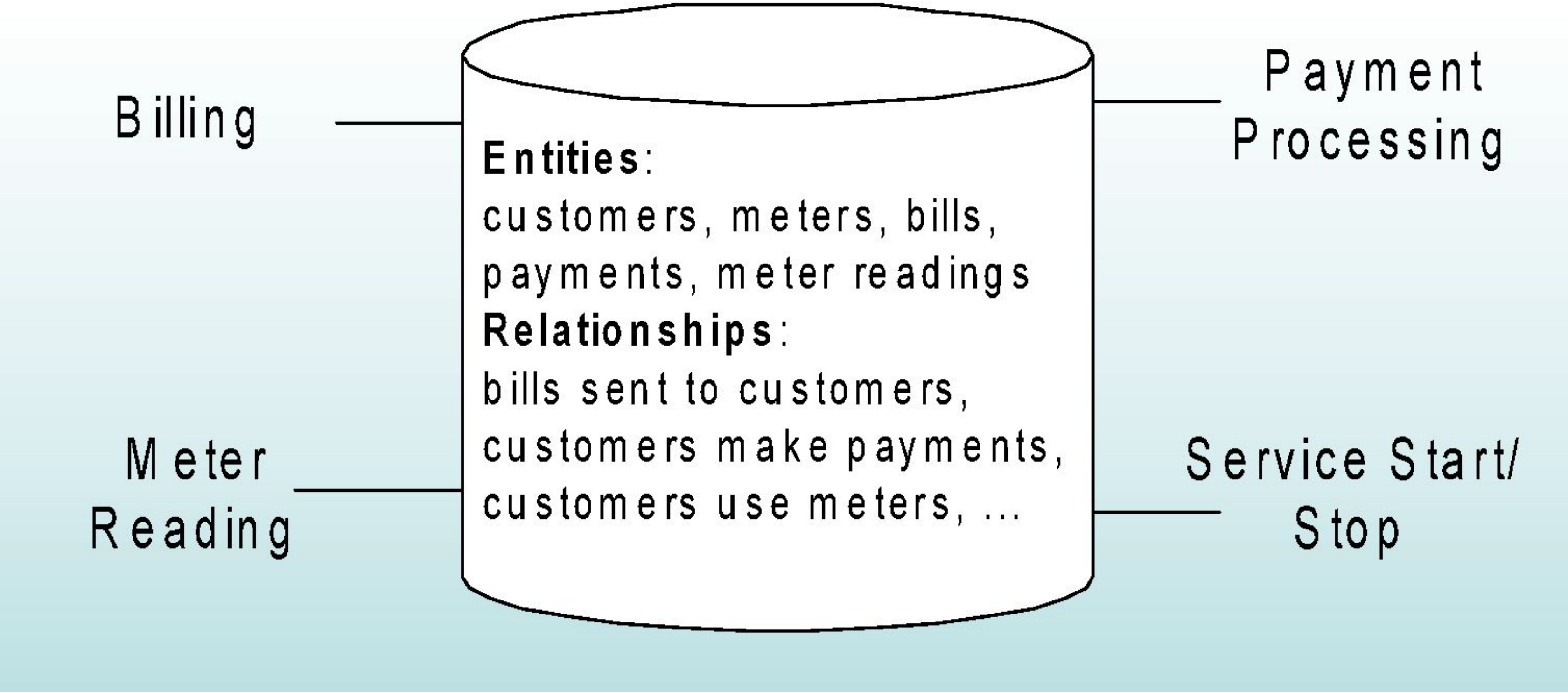
Primary key **Foreign key** **Primary key**

Relational Database Terminology



EMPLOYEE_ID	LAST_NAME	FIRST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID
100	King	Steven	24000		90
101	Kochhar	Neena	17000		90
102	De Haan	Lex	17000		90
103	Hunold	Alexander	9000		60
104	Ernst	Bruce	6000		60
107	Lorentz	Diana	4200		60
124	Mourgos	Kevin	5800		50
141	Rajs	Trenna	3500		50
142	Davies	Curtis	3100		50
143	Matos	Randall	2600		50
144	Vargas	Peter	2500		50
149	Zlotkey	Eleni	10500	.2	80
174	Abel	Ellen	11000	.3	80
176	Taylor	Jonathon	8600	.2	80
178	Grant	Kimberely	7000	.15	
200	Whalen	Jennifer	4400		10
201	Hartstein	Michael	13000		20
202	Fay	Pat	6000		20
205	Higgins	Shelley	12000		110
206	Gietz	William	8300		110

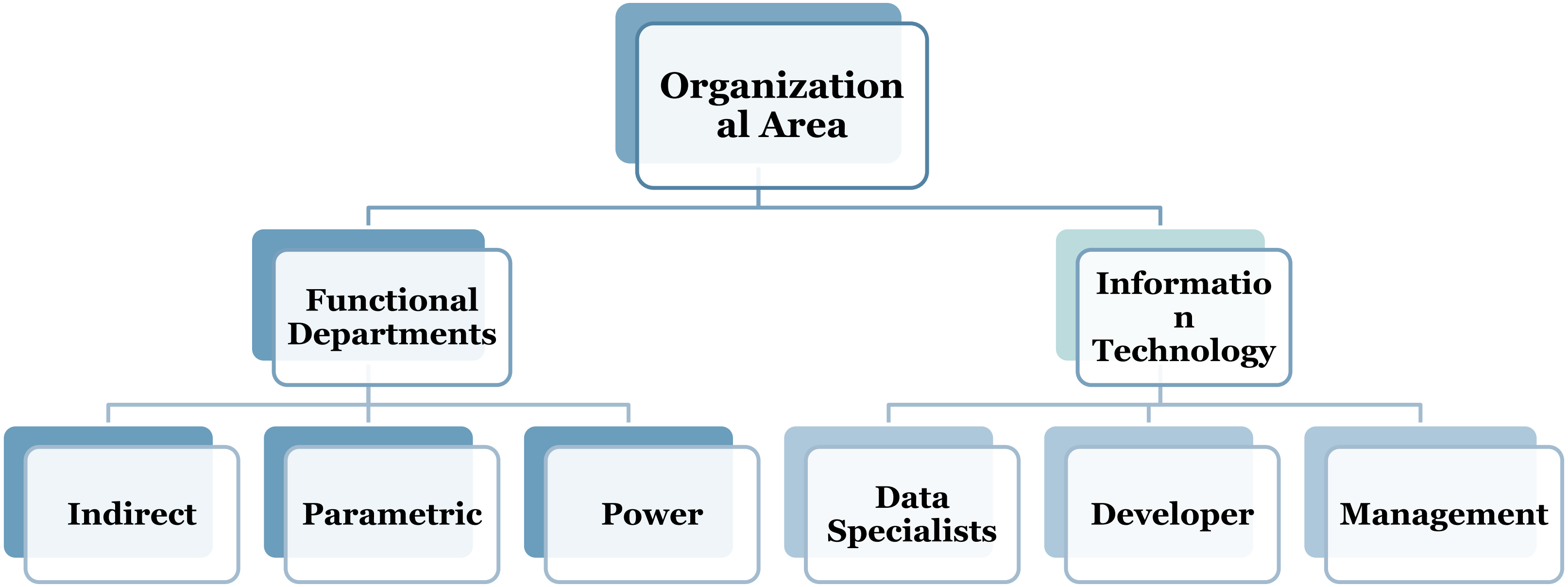
Water Utility Database



Entities:
customers, meters, bills, payments, meter readings

Relationships:
bills sent to customers, customers make payments, customers use meters, …

Organizational Roles



Organizational Area

Functional Departments

Information Technology

Indirect

Parametric

Power

Data Specialists

Developer

Management

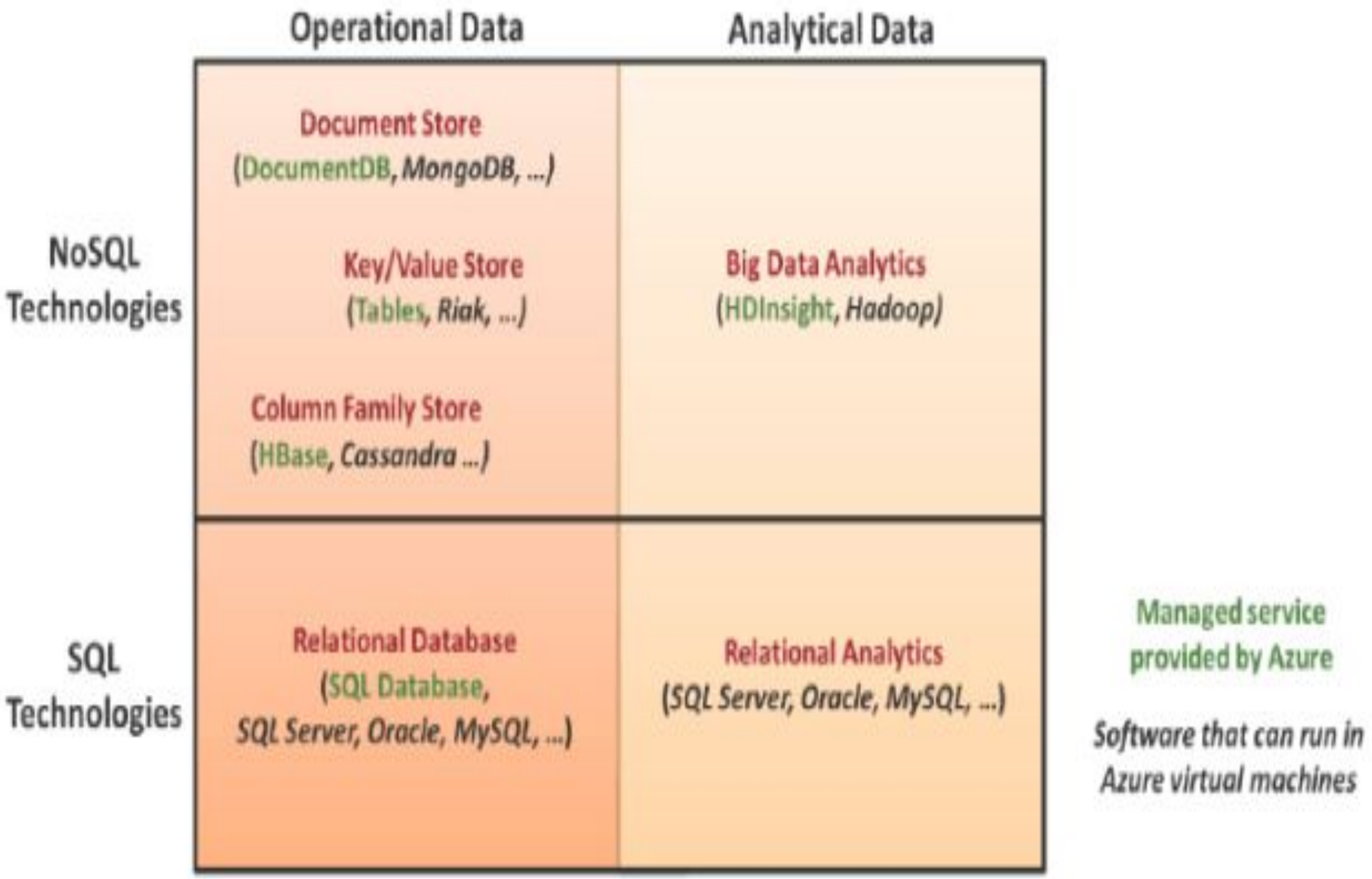
Database Specialists

- Database administrator (DBA)
  - More technical
  - DBMS specific skills
- Data administrator
  - Less technical
  - Planning role

Should I be using NoSQL/Relational Databases?

- NoSQL Data storage systems makes sense for applications that need to deal with very large semi-structured data
  - Log Analysis
  - Social Networking Feeds

- Most of us work on organizational databases, which are not that much large and have low or may regular update/query rates
  - regular relational databases are the correct solution for such applications



	Operational Data	Analytical Data
NoSQL Technologies	Document Store (DocumentDB, *MongoDB, ...)* Key/Value Store (Tables, *Riak, ...)* Column Family Store (HBase, *Cassandra ...)*	Big Data Analytics (HDInsight, *Hadoop)*
SQL Technologies	Relational Database (SQL Database, *SQL Server, Oracle, MySQL, ...*)	Relational Analytics (*SQL Server, Oracle, MySQL, ...*)

Managed service provided by Azure

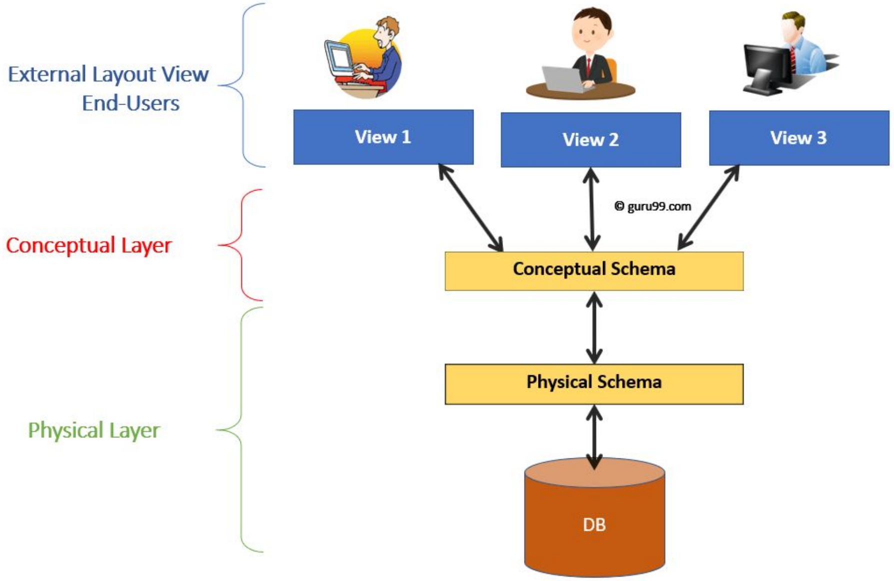
Software that can run in Azure virtual machines

Planning and Organizing a Database

- Define the purpose of the database.
- Define the type of the database.
- Outline a database architectural design.
- Choose the database name.

- Conceptual Level

□ Mainly used for requirement analysis phase (ER-Diagram)



- Representation level

□ Choose a database model (RDBMS)

- Physical level

□ Built-In Database Objects

Relational Database Management System

An RDBMS moves data into a database, stores the data, and retrieves it so that it can be manipulated by applications.

- Logical operations
- Physical operations

Why is the relational data model commercially dominant?

Here are some examples of popular DBMS used these days:

- Oracle
- SQL Server
- IBM DB2
- PostgreSQL
- Amazon SimpleDB (cloud based)
- MySql etc.

Database Design (Schema)

- Covering database design with multiple tables, foreign keys, and the JOIN operation, which build a Physical Data Schema.

A database schema can be divided broadly into three categories

- **Physical Database Schema** – This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.

- **Logical Database Schema** – This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

- **View Schema-** This generally describes end user interaction with database systems.

Physical Data Schema..(Built-In Database Objects..)

- **Data dictionary**: Contains descriptions of the objects in the database

- **Dynamic performance tables**: Contains information used by the database administrator (DBA) to monitor and tune the database and instance

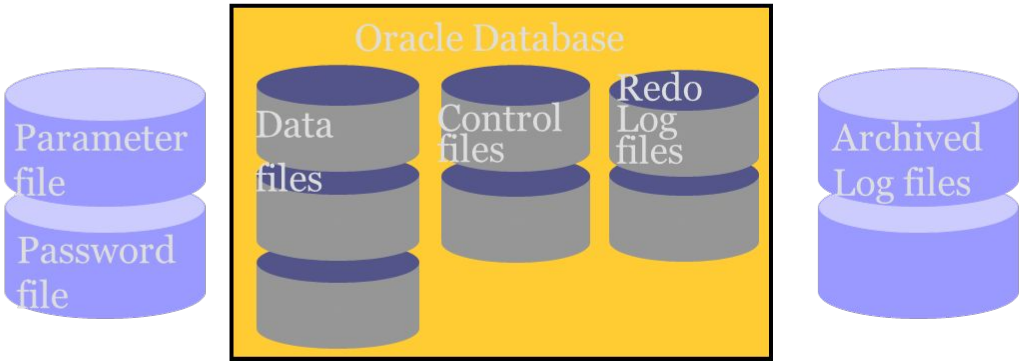
- **Database event triggers**: Triggers are procedures that execute implicitly whenever a table or view is modified, or when some user actions or database system actions occur

Data Dictionary Content

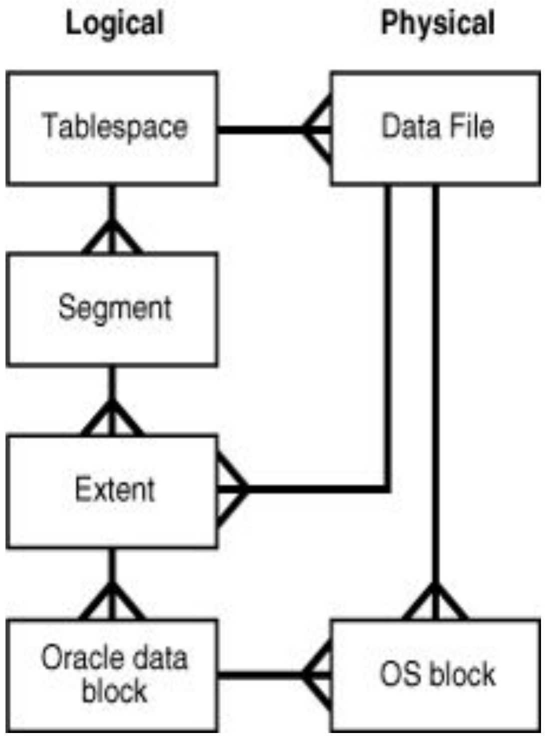
- Logical and Physical Database structures
- Definition and space allocation of objects
- Integrity constraints
- Users
- Roles
- Privileges

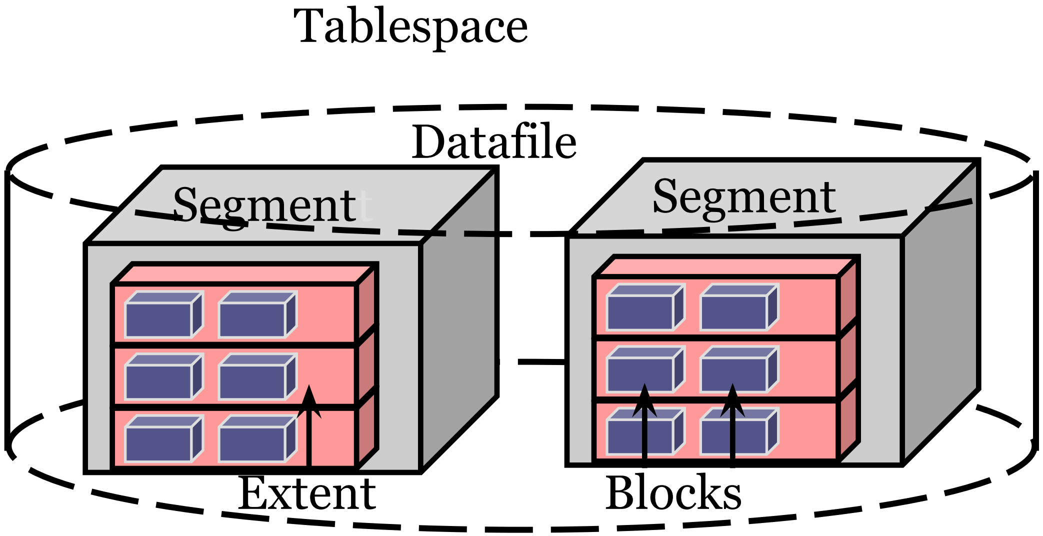
Oracle Database

- An Oracle database consists of operating system files, also known as database files, that provide the actual physical storage for database information.



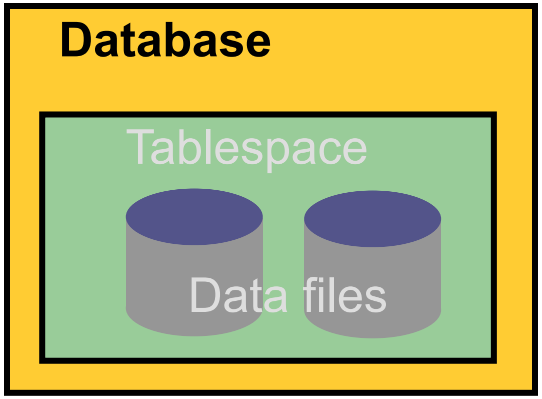
Logical Structure





Tablespaces and Data Files

- Oracle stores data logically in tablespace and physically in data files.
- Can belongs to one database at a time
- Consists of one or more data files



Data files:

- Can belong to only one tablespace and one database
- Are a repository for schema object data

Types of Tablespace

- System tablespace

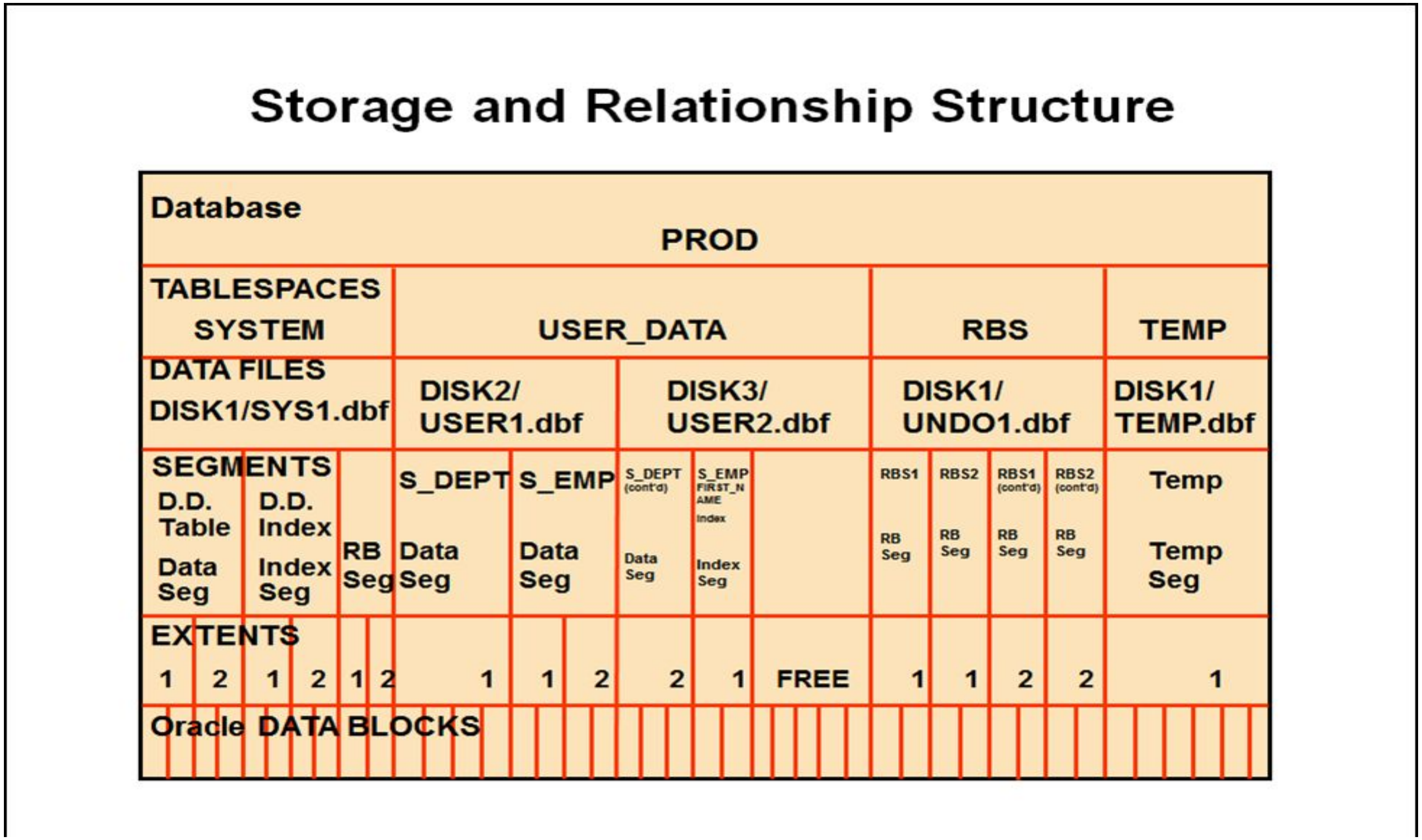
1. Created with the database
2. Required in all databases
3. Contains the data dictionary,
4. Should not contain user data, although it is allowed

- Non-system tablespace

1. Control the amount of space allocated to the user’s objects
2. Enable more flexibility in database administration

Creating Tablespace

Create Tablespace userdata DATAFILE '/u01/oradata/userdata01.db' SIZE 10G



Storage and Relationship Structure

Database
PROD

TABLESPACES
SYSTEM
USER_DATA
RBS
TEMP

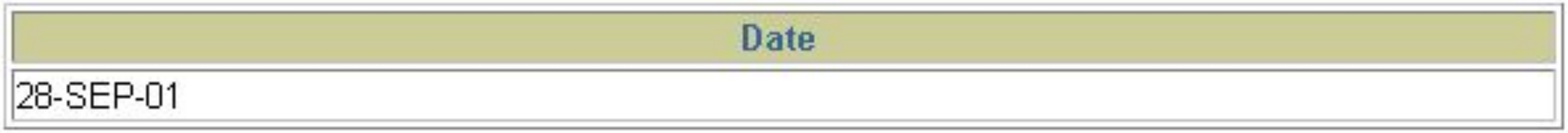
DATA FILES
DISK1/SYS1.dbf
DISK2/ USER1.dbf
DISK3/ USER2.dbf
DISK1/ UNDO1.dbf
DISK1/ TEMP.dbf

SEGMENTS
D.D. Table Data Seg
D.D. Index Index Seg
RB Seg
S_DEPT Data Seg
S_EMP Data Seg
S_DEPT (cont'd) Data Seg
S_EMP FIRST_NAME Index Index Seg
RBS1 RB Seg
RBS2 RB Seg
RBS1 (cont'd) RB Seg
RBS2 (cont'd) RB Seg
Temp Temp Seg

EXTENTS
1 2 1 2 1 2 1 1 2 2 1 FREE 1 1 2 2 1

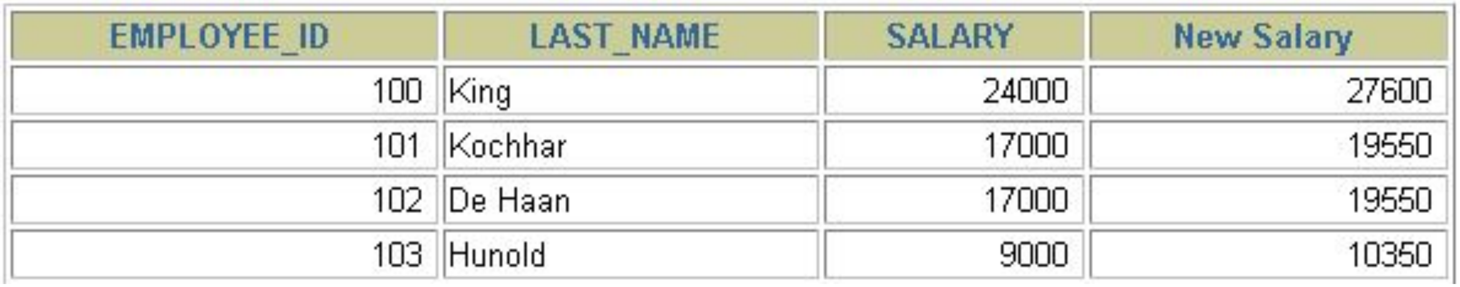
Oracle DATA BLOCKS

Write a query to display the current date. Label the column `Date`.

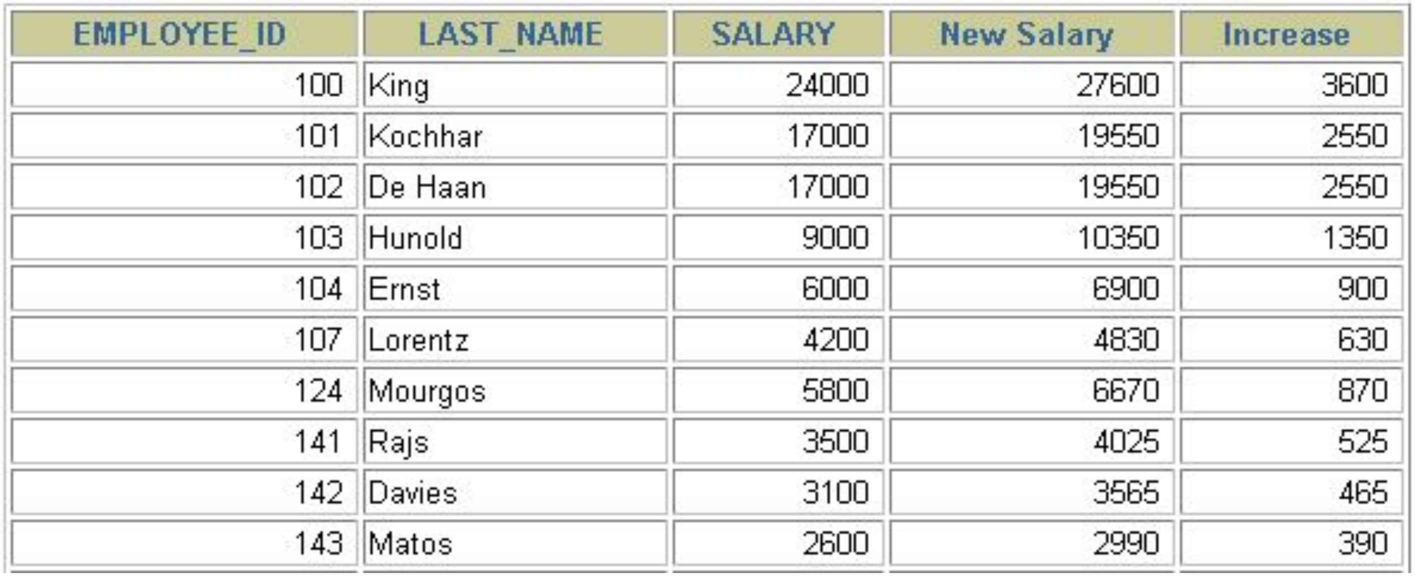


Date
28-SEP-01

For each employee, display the employee number, last_name, total_salary, and salary increased by 15% and expressed as a whole number. Label the column `New Salary`

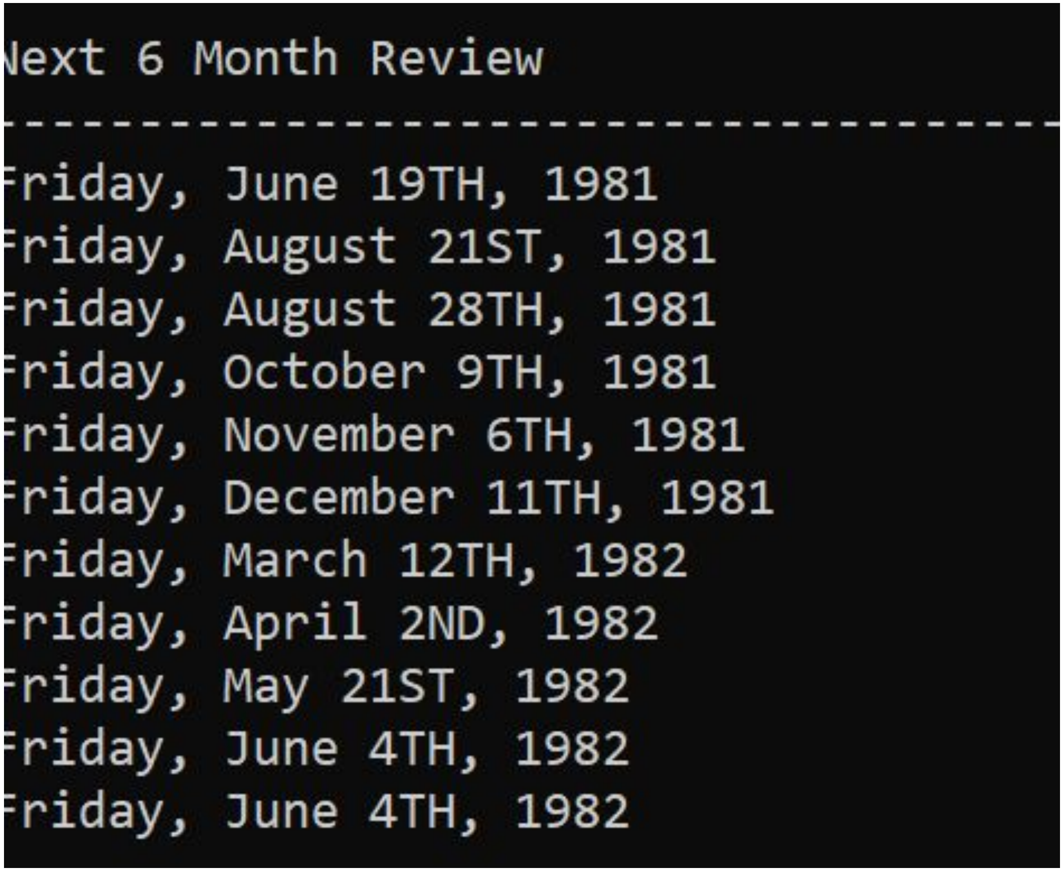


EMPLOYEE_ID	LAST_NAME	SALARY	New Salary
100	King	24000	27600
101	Kochhar	17000	19550
102	De Haan	17000	19550
103	Hunold	9000	10350



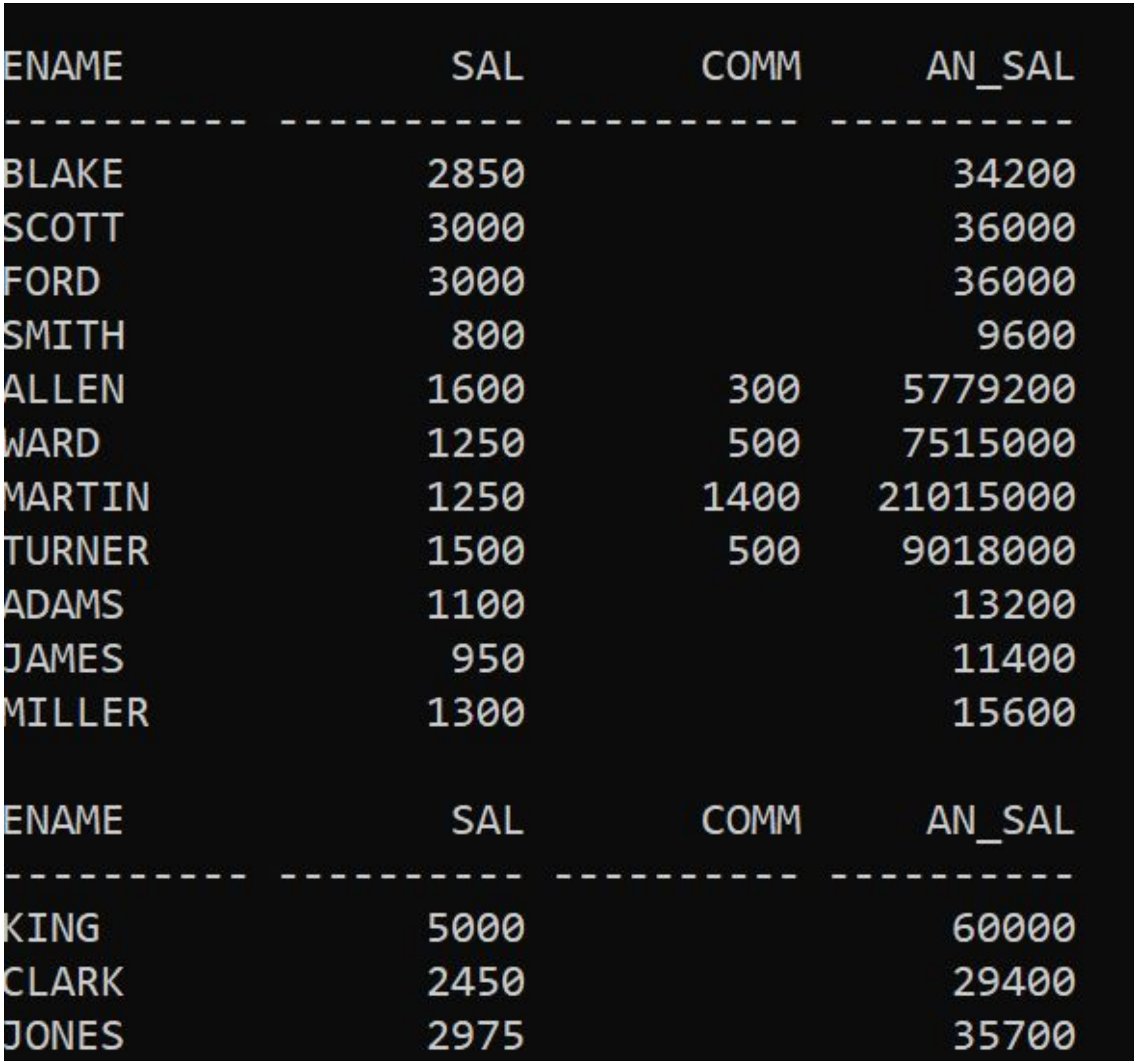
EMPLOYEE_ID	LAST_NAME	SALARY	New Salary	Increase
100	King	24000	27600	3600
101	Kochhar	17000	19550	2550
102	De Haan	17000	19550	2550
103	Hunold	9000	10350	1350
104	Ernst	6000	6900	900
107	Lorentz	4200	4830	630
124	Mourgos	5800	6670	870
141	Rajs	3500	4025	525
142	Davies	3100	3565	465
143	Matos	2600	2990	390

Display the date of the next Friday that is six months from the hire date. The resulting date should appear as Friday, August 13th, 1999. Order the results by hire date.



```
Next 6 Month Review
-------------------------------------
Friday, June 19TH, 1981
Friday, August 21ST, 1981
Friday, August 28TH, 1981
Friday, October 9TH, 1981
Friday, November 6TH, 1981
Friday, December 11TH, 1981
Friday, March 12TH, 1982
Friday, April 2ND, 1982
Friday, May 21ST, 1982
Friday, June 4TH, 1982
Friday, June 4TH, 1982
```

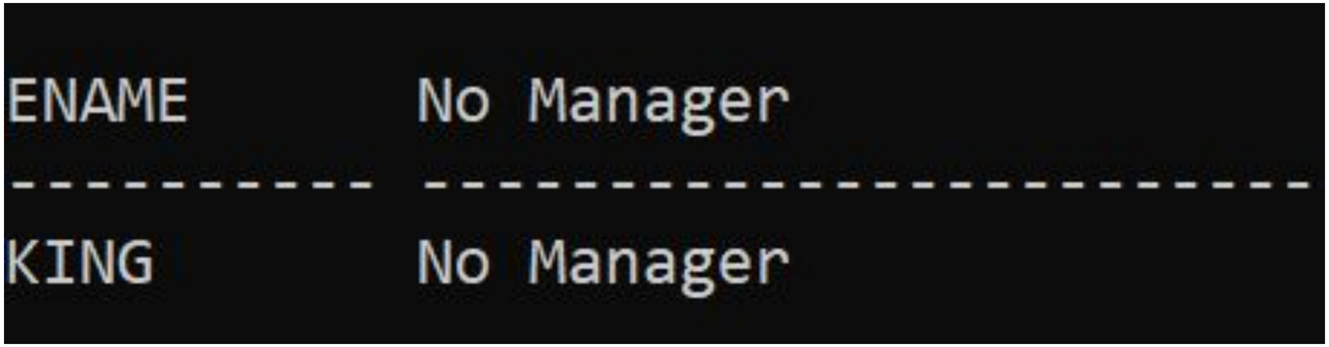
```
 SELECT   TO_CHAR(NEXT_DAY(ADD_MONTHS
(hiredate, 6), 'FRIDAY')  'fmDay, Month DDth, YYYY')
"Next 6 Month Review"  FROM      emp   ORDER BY
hiredate
```



```
ENAME             SAL       COMM     AN_SAL
---------- ---------- ---------- ----------
BLAKE            2850                 34200
SCOTT            3000                 36000
FORD             3000                 36000
SMITH             800                  9600
ALLEN            1600        300    5779200
WARD             1250        500    7515000
MARTIN           1250       1400   21015000
TURNER           1500        500    9018000
ADAMS            1100                 13200
JAMES             950                 11400
MILLER           1300                 15600

ENAME             SAL       COMM     AN_SAL
---------- ---------- ---------- ----------
KING             5000                 60000
CLARK            2450                 29400
JONES            2975                 35700
```

```
 SELECT ename, sal, comm,  (sal*12) +
(sal*12*nvl(comm,0)) AN_SAL   FROM   emp
```



```
ENAME      No Manager
---------- ------------------------
KING       No Manager
```